Klasifikasi Sentimen Emosi pada Text Pesan Singkat

Waffiq Maaroja   
*Departemen Ilmu Komputer dan Elektronika  
Universitas Gadjah Mada*Yogyakarta, Indonesia  
waffiqmaaroja@mail.ugm.ac.id

Muhammad Salam  
*Departemen Ilmu Komputer dan Elektronika  
Universitas Gadjah Mada*Yogyakarta, Indonesia  
muhammadsalam1998@mail.ugm.ac.id

Nisrina Hanifa Setiono  
*Departemen Ilmu Komputer dan Elektronika  
Universitas Gadjah Mada*Yogyakarta, Indonesia  
nisrinahanifasetiono@mail.ugm.ac.id

*Abstrak*— Analisis emosi pada text pesan singkat bertujuan untuk melakukan klasifikasi teks berdasarkan kategori sentimen emosi guna membantu mengidentifikasi berbagai tindakan dan kejadian yang berhubungan dengan manusia. Studi ini memanfaatkan dataset Emotions yang terdiri dari teks pesan singkat dari Twitter yang dikategorikan ke dalam enam jenis emosi: sadness, joy, love, anger, fear, dan surprise. Proses analisis melibatkan pra-pemrosesan data, ekstraksi fitur, dan pelatihan model klasifikasi menggunakan Logistic Regression. Studi ini berfokus pada eksplorasi metode ekstraksi fitur TF-IDF, banyak komponen fitur yang dipilih, dan pengaruh bobot kelas dalam memaksimalkan klasifikasi teks. Hasil eksperimental menunjukkan bahwa model yang dilatih menggunakan fitur TF-IDF dan bobot kelas same dengan banyak komponen fitur sebesar 6% dari keseluruhan kosakata serta memiliki performa yang lebih baik.

Kata kunci—analisis sentimen, teks, machine learning, natural language text (NLP), logistic regression, tf-idf.

# Pendahuluan

Di era digital saat ini, interaksi manusia seringkali terjadi melalui berbagai platform media sosial, terutama Twitter. Sebagai platform microblogging, Twitter menjadi sumber besar data teks yang tidak hanya mengandung informasi tekstual tetapi juga penyampaian emosi. Kemampuan untuk memahami dan menganalisis konten ini secara otomatis membuka peluang di bidang pemrosesan bahasa alami (NLP). Analisis emosi dan sentimen dari data tekstual dapat membantu mengidentifikasi berbagai tindakan dan kejadian yang berhubungan dengan manusia, seperti menghitung tingkat kepuasan hidup, berbagai parameter pelanggan, kesejahteraan masyarakat di suatu negara, dan mencegah potensi upaya yang merugikan diri sendiri maupun orang lain melalui teks media sosial [1] [2].

Studi ini bertujuan untuk melakukan klasifikasi teks berdasarkan beberapa kategori sentimen emosi bernuansa negatif, netral, maupun positif [4]. Klasifikasi ini akan dikembangkan menggunakan metode machine learning logistic regression, yang merupakan klasifikasi statistik untuk memprediksi kemungkinan kejadian dengan menganalisis hubungan antara satu set variabel independen dan variabel dependen biner atau multinomial, berdasarkan dataset yang bersumber dari Twitter. Untuk mencapai hasil akurat dalam klasifikasi, studi ini juga akan menerapkan metode ekstraksi fitur tertentu dan model pembelajaran mesin yang mendukung proses klasifikasi pada banyak kelas. Analisa lebih lanjut dilakukan pada proses ekstraksi fitur untuk mengetahui efektivitas atau pengaruhnya terhadap hasil klasifikasi.

# Dataset

Dataset yang digunakan pada studi ini adalah dataset Emotions [3] dimana tiap entri terdiri dari sebuah segmen teks yang mewakili pesan Twitter berbahasa inggris dan label yang menunjukkan emosi dominan yang disampaikan. Dataset berisi sebanyak 416809 teks yang masing-masing dikategorikan ke 6 jenis emosi yaitu “*sadness*”, “*joy*”, “*love*”, “*anger*”, “*fear*”, dan “*surprise*”. Contoh teks pada masing-masing kategori ditunjukkan pada Tabel I serta distribusi dataset pada masing-masing kategori ditunjukkan pada Gambar 1.

1. Teks Sampel Tiap Kelas

| Teks | Label |
| --- | --- |
| *i can still lose the weight without feeling deprived* | *sadness* |
| *i feel my portfolio demonstrates how eager i am to learn but some who know me better might call it annoyingly persistent* | *joy* |
| *i think a guy can make up for lacking funds in romantic ways to make his lady feel taken care of and loved* | *love* |
| *i need to feel dangerous and pretty so here a striking dance pick deep in vogue minutes ago* | *anger* |
| *i dunno i just feel scared to walk in after awhile it got ok but when i walk into this room w poorly erected brick walls separating the room into xm partitions* | *fear* |
| *im feeling a bit amazed and grateful about having landed amid such a congregation* | *suprise* |

A graph with blue bars

Description automatically generated

1. Distribusi data pada setiap kategori/kelas

Dari distribusi data tersebut, diketahui bahwa dataset tersebut lebih banyak mengandung teks dengan emosi positif ("*joy*") dan negatif ("*sadness*"). Emosi lainnya seperti "*anger*", "*fear*", "*love*", dan "*surprise*" memiliki frekuensi yang lebih rendah dengan frekuensi dibawah setengah dari frekuensi kelas mayoritas, dimana "*surprise*" menjadi kategori paling jarang ditemukan. Hal ini dapat mempengaruhi keseimbangan dalam model machine learning yang akan dikembangkan sehingga diperlukan teknik penanganan data yang tidak seimbang (*imbalanced data*) tersebut.

# Metodologi

A diagram of a process

Description automatically generated

1. Alur pelatihan model

Alur pelatihan model yang telah ditunjukkan pada Gambar 2 terdiri dari tahap pra-pemrosesan data, ekstraksi fitur, dan pelatihan model klasifikasi.

## Pra-pemrosesan Data

Dalam analisis dataset yang digunakan, terdapat sebanyak 22817 teks yang muncul lebih dari sekali dalam dataset. Oleh karena itu, untuk menjaga keunikan setiap entri, semua duplikasi dihilangkan sehingga setiap teks hanya muncul tepat sekali. Proses selanjutnya adalah *cleansing*, yang bertujuan untuk menghilangkan karakter-karakter yang tidak penting dalam teks. Beberapa tahapan *cleansing* meliputi:

* Menghapus tanda baca seperti titik, koma, tanda tanya, dan tanda seru dihapus untuk menyederhanakan teks.
* Menghapus semua angka di dalam teks karena biasanya tidak relevan dalam analisis emosi.
* Menghapus karakter khusus seperti @, #, %, dan simbol lainnya dihilangkan untuk membersihkan teks dari elemen-elemen yang tidak perlu.
* Menghapus spasi berlebih antara kata-kata untuk menjaga konsistensi dalam teks.

Setelah proses *cleansing*, dilakukan *case folding*, yaitu mengubah semua karakter menjadi huruf kecil untuk menstandarisasi teks, sehingga perbedaan antara huruf besar dan kecil tidak mempengaruhi analisis.

Tahap berikutnya adalah *chatword expanding*, yaitu mengganti singkatan dan slang menjadi bentuk lengkapnya. Contohnya, "*u*" menjadi "*you*" dan "*r*" menjadi "*are*". Proses ini bertujuan untuk meningkatkan kejelasan dan makna teks.

Selanjutnya, dilakukan *stopword removal* dimana kata-kata yang dianggap tidak memberikan informasi penting atau relevan, seperti "*the*", "*is*", "*in*", dan sejenisnya dihapus dari teks.

Akhirnya, proses *stemming* diterapkan, yang mengubah kata-kata menjadi bentuk dasar atau akar katanya. Misalnya, kata "*running*" diubah menjadi "*run*". Proses ini membantu dalam menyederhanakan kata-kata sehingga analisis teks menjadi lebih efisien. Dengan menerapkan tahapan-tahapan ini, data teks menjadi lebih bersih, terstandarisasi, dan siap untuk dianalisis lebih lanjut.

## Ekstraksi Fitur

Dalam proses analisis teks, representasi numerik dari teks diperlukan agar data dapat digunakan pada tahapan selanjutnya, termasuk proses pembelajaran model yang digunakan untuk melakukan klasifikasi. Beberapa fitur yang dapat digunakan adalah Term Frequency (TF) dan Inverse Document Frequency (IDF).

TF mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen, sedangkan IDF mengukur seberapa penting sebuah kata dalam keseluruhan korpus [5]. Kata-kata yang sering muncul di banyak dokumen memiliki IDF rendah, sedangkan kata-kata yang jarang muncul memiliki IDF tinggi. TF dari sebuah kata pada dokumen tertentu ditunjukkan pada persamaan (1). Sedangkan IDF dari sebuah kata dihitung dengan rumus logaritmik dan ditunjukkan pada persamaan (2). Selain itu, TF dan IDF dapat digabungkan untuk membentuk fitur TF-IDF yang memberikan bobot lebih tinggi pada kata-kata yang sering muncul dalam dokumen tertentu tetapi jarang muncul di dokumen lain. TF-IDF dihitung dengan mengalikan nilai TF dengan nilai IDF.





Fitur yang akan digunakan pada studi ini adalah TF saja dan TF-IDF. Kedua fitur akan dibandingkan untuk mengetahui seberapa signifikan perbedaan pada hasil klasifikasi. Sebelum fitur digunakan, setiap fitur akan dinormalisasi sehingga vektor tersebut menjadi vektor unit (vektor dengan panjang 1). Hal ini dilakukan agar model klasifikasi dapat mempelajari setiap komponen pada fitur dengan baik.

## Model Klasifikasi

*Logistic Regression* adalah model statistik yang sering digunakan untuk masalah klasifikasi biner, di mana tujuan utamanya adalah untuk memprediksi probabilitas kejadian dari dua kategori [6]. Model ini bekerja dengan menggunakan fungsi logit untuk mengestimasikan hubungan antara satu atau lebih variabel independen dengan variabel dependen kategori.

Dalam studi ini, model digunakan untuk mengklasifikasikan emosi ke dalam lebih dari dua kategori, sehingga digunakan *Multinomial Logistic Regression*. Model tersebut adalah generalisasi dari *Logistic Regression* yang memungkinkan klasifikasi ke dalam lebih dari dua kelas. Dalam model ini, satu kelas dipilih sebagai referensi, dan probabilitas kelas lainnya dihitung relatif terhadap kelas referensi tersebut.

Misalkan terdapat sebanyak kelas. Untuk data ke- dengan fitur , maka probabilitas data ke- merupakan kelas ke- adalah yang ditunjukkan pada persamaan (3) dan (4) berikut





Bobot pada model tersebut adalah yang berupa matriks dan dipilih sedemikian sehingga nilai *loss* pada persamaan (5) berikut bernilai minimum.





dimana merupakan label dari data ke-dan merupakan bobot kelas ke-. Pada analisis ini, terdapat 2 skenario untuk menentukan bobot kelas yang digunakan, yaitu bernilai 1 untuk semua kelas atau memiliki nilai yang berbanding terbalik dengan frekuensi data pada kelas tersebut yang dirumuskan pada persamaan (7).





Skenario kedua digunakan untuk mengatasi permasalahan *data imbalance* dimana perubahan nilai *loss* akan lebih sensitif terhadap kelas dengan frekuensi terkecil. Sedangkan nilai adalah invers dari faktor regularisasi pada fungsi *loss* yang berupa konstanta dimana semakin kecil nilainya maka pengaruh regularisasi akan semakin besar. Nilai yang digunakan pada analisis ini adalah 1.

Pada tahap pelatihan, proses pencarian nilai minimum dari nilai *loss* akan menggunakan metode optimasi yaitu algoritma *gradient descent* untuk memperbarui bobot model. Pada analisis ini, metode *gradient descent* yang digunakan adalah SAGA [7]. Proses pelatihan akan berhenti ketika nilai *loss* sudah konvergen atau iterasi sudah mencapai nilai tertentu, yaitu 1000 iterasi.

# Hasil dan Analisis

Implementasi dilakukan menggunakan Google Colab dengan spesifikasi CPU Intel Xeon dengan 2 vCPU (virtual CPU) @ 2.2GHz dan RAM sebesar 13GB. Waktu yang diperlukan pada tahap pelatihan dianalisa untuk mengetahui dampak setiap skenario percobaan dari sisi waktu pemrosesan.

Sebelum melakukan proses pelatihan dan pengujian, dataset akan dibagi menjadi 2 bagian yaitu data latih dan data uji dengan perbandingan 4:1. Setelah melalui tahap pra-pemrosesan data, banyak kosakata unik pada data latih adalah sebanyak 46082 kata, yang merepresentasikan banyak fitur. Namun pada percobaan ini, banyak fitur akan dibatasi sehingga hanya akan dipilih sebagian kosakata dengan frekuensi terbanyak pada corpus data latih. Waktu pemrosesan ekstraksi fitur untuk setiap skenario banyak komponen fitur yang diambil ditunjukkan pada Gambar 3.

A graph of blue lines

Description automatically generated

1. Waktu pemrosesan ekstraksi fitur

Berdasarkan grafik yang ditunjukkan pada Gambar 3 dapat diketahui bahwa pembatasan banyak komponen pada fitur atau pembatasan kosakata tidak berpengaruh terhadap lama waktu pemrosesan proses ekstraksi fitur. Hal ini disebabkan karena setiap kosakata tanpa terkecuali akan dihitung frekuensinya, kemudian dilakukan pengurutan semua kosakata berdasarkan frekuensi dan barulah diambil komponen fitur yang diperlukan, sehingga kompleksitas waktu ekstraksi fitur tidak berkaitan dengan banyak komponen fitur yang diambil, namun akan berpengaruh terhadap kompleksitas ruang atau memori yang terpakai.

Terdapat 2 skenario fitur yang digunakan, yaitu fitur TF saja dan fitur TF-IDF. Waktu yang diperlukan untuk melakukan ekstraksi fitur pada data latih untuk setiap skenario ditunjukkan pada Tabel II.

1. Perbandingan Waktu Ekstraksi Fitur

| Fitur | Rata-rata Waktu (dalam detik) |
| --- | --- |
| TF | 7.324586200714111 |
| TF-IDF | 7.409243779182434 |

Berdasarkan Tabel II dapat diketahui bahwa waktu yang diperlukan untuk mengekstrak fitur TF-IDF tidak berubah secara signifkan, yaitu mengalami kenaikan sekitar 1% jika dibandingkan dengan ekstraksi fitur TF saja, sehingga pemilihan jenis fitur yang digunakan tidak begitu berpengaruh terhadap waktu pemrosesan secara keseluruhan.

Pada tahap pelatihan, terdapat 2 skenario bobot kelas pada perhitungan nilai *loss* model klasifikasi yang digunakan, yaitu bobot kelas *same* dimana untuk setiap kelas memiliki bobot 1 dan bobot kelas *balanced* dimana nilai bobot berbanding terbalik dengan frekuensi data pada masing-masing kelas sesuai persamaan (7). Waktu yang diperlukan untuk melatih model pada setiap skenario ditunjukkan pada Gambar 4.

A graph with blue lines

Description automatically generated

(a) Bobot kelas *same*

A graph with red lines

Description automatically generated

(b) Bobot kelas *balanced*

1. Perbandingan waktu yang diperlukan pada pelatihan model untuk masng-masing skenario bobot kelas

Berdasarkan Gambar 4 dapat diketahui bahwa waktu pelatihan model berbanding lurus dengan banyak komponen fitur yang digunakan, sehingga semakin banyak komponen fitur yang digunakan maka akan semakin lama proses pelatihan model yang diperlukan. Disisi lain, penggunaan bobot kelas *balanced* dapat memperlambat waktu pelatihan secara sangat signifikan jika dibandingan dengan bobot kelas *same*. Hal ini disebabkan karena nilai *loss* pada model dengan bobot kelas *balanced* lebih sulit mencapai konvergensi dan bahkan masih belum mampu konvergen hingga mencapai batas iterasi maksimal, yaitu 1000 iterasi. Sebaliknya, nilai *loss* pada model dengan bobot kelas *same* mampu konvergen secara cepat sebelum mencapai iterasi maksimal.

Pada tahap pengujian, performa model akan dievaluasi menggunakan data uji dengan beberapa metrik yaitu akurasi, *precision*, *recall*, dan skor F1. Pada kasus klasifikasi dengan lebih dari 2 label, akurasi (ACC) dihitung dengan membandingkan banyak prediksi yang sesuai dengan banyak data keseluruhan, sedangkan perhitungan *precision*, *recall*, dan skor F1 dihitung pada masing-masing kelas kemudian diambil nilai rata-ratanya, yang biasa disebut sebagai *macro average* *precision* (MAP), *macro average* *recall* (MAR), dan *macro average* F1 (MAF). Skenario model dengan bobot kelas dan fitur yang digunakan ditunjukkan pada Tabel III serta perbandingan nilai metrik pada setiap skenario model dan skenario banyak komponen fitur (rasio terhadap keseluruhan banyak komponen) yang digunakan ditunjukkan pada Tabel IV.

1. Skenario Model

| Model | Fitur | Bobot Kelas |
| --- | --- | --- |
| LR-TF | TF | *same* |
| LRB-TF | TF | *balanced* |
| LR-TFIDF | TF-IDF | *same* |
| LRB-TFIDF | TF-IDF | *balanced* |

1. Perbandingan Performa Model

| Rasio | Metrik | LR-TF | LRB-TF | LR-TFIDF | LRB-TFIDF |
| --- | --- | --- | --- | --- | --- |
| 2% | ACC | 0.871 | 0.860 | **0.872** | 0.861 |
| MAP | 0.853 | 0.798 | **0.854** | 0.796 |
| MAR | 0.817 | **0.864** | 0.818 | 0.864 |
| MAF | 0.834 | 0.824 | **0.835** | 0.823 |
| 30% | ACC | 0.905 | 0.894 | **0.906** | 0.891 |
| MAP | 0.879 | 0.860 | **0.882** | 0.826 |
| MAR | 0.864 | 0.866 | 0.863 | **0.909** |
| MAF | 0.871 | 0.859 | **0.872** | 0.858 |
| 60% | ACC | 0.905 | 0.894 | **0.906** | 0.896 |
| MAP | 0.879 | 0.834 | **0.882** | 0.839 |
| MAR | 0.864 | **0.906** | 0.862 | 0.901 |
| MAF | 0.871 | 0.861 | **0.871** | 0.865 |
| 100% | ACC | 0.905 | 0.894 | **0.906** | 0.882 |
| MAP | 0.879 | 0.835 | **0.882** | 0.841 |
| MAR | 0.864 | **0.907** | 0.862 | 0.868 |
| MAF | 0.871 | 0.864 | **0.871** | 0.848 |

A graph of different colored lines

Description automatically generated

1. Performa model LR-TFIDF

Berdasarkan Tabel IV dapat diketahui bahwa dari sisi bobot kelas yang digunakan, model yang menggunakan bobot kelas *same* memiliki performa yang sedikit lebih baik daripada model yang menggunakan bobot kelas *balanced*. Sedangkan dari sisi pemilihan fitur, model yang dilatih menggunakan fitur TF saja memiliki performa yang setara dengan model yang dilatih menggunakan fitur TF-IDF. Secara keseluruhan, model LR-TFIDF memiliki performa terbaik dibandingkan dengan model lainnya. Selain itu berdasarkan Gambar 5 dapat diketahui bahwa banyak komponen fitur yang optimal adalah sekitar 6% dari keseluruhan komponen fitur atau kosakata, dimana selebihnya tidak mempengaruhi performa model, sehingga dengan membatasi banyak komponen pada fitur maka proses pelatihan akan lebih cepat namun tetap memberikan performa model yang baik.

# Kesimpulan

Dari hasil analisis ini dapat disimpulkan bahwa waktu pemrosesan ekstraksi fitur tidak dipengaruhi oleh jumlah komponen fitur yang dipilih, melainkan lebih dipengaruhi oleh kompleksitas ruang atau memori yang digunakan. Meskipun penambahan fitur TF-IDF meningkatkan waktu ekstraksi fitur sekitar 1%, pengaruhnya terhadap waktu pemrosesan secara keseluruhan tidak signifikan. Pada tahap pelatihan, jumlah komponen fitur yang digunakan secara langsung mempengaruhi waktu pelatihan, dengan lebih banyak komponen fitur membutuhkan waktu pelatihan yang lebih lama. Selain itu, penggunaan bobot kelas *balanced* memperlambat waktu pelatihan secara signifikan karena kesulitan mencapai konvergensi dibandingkan dengan bobot kelas *same*.

Pada tahap pengujian, model dengan bobot kelas *same* menunjukkan performa yang sedikit lebih baik dibandingkan dengan model menggunakan bobot kelas *balanced*. Dalam hal pemilihan fitur, model yang dilatih dengan fitur TF menunjukkan performa yang setara dengan model yang dilatih menggunakan fitur TF-IDF, dengan model LR-TFIDF memiliki performa terbaik secara keseluruhan. Jumlah komponen fitur yang optimal ditemukan sekitar 6% dari keseluruhan kosakata, di mana membatasi jumlah fitur ini dapat mempercepat proses pelatihan tanpa mengorbankan performa model.

Secara keseluruhan, hasil analisis ini memberikan wawasan berharga dalam optimalisasi pemilihan fitur dan penggunaan bobot kelas untuk mencapai efisiensi waktu dan performa yang baik dalam model klasifikasi teks menggunakan *Logistic Regression*.

Referensi

1. Y. Wang, J. Guo, C.Yuan, and B. Li, "Sentiment Analysis of Twitter Data," *Applied Sciences,* vol.12, no. 22, 2022.
2. N. Parveen, P. Chakrabarti, B. T. Hung, and A. Shaik, "Twitter sentiment analysis using hybrid gated attention recurrent network," *J Big Data*, vol. 10, no. 50, 2023. Available: <https://doi.org/10.1186/s40537-023-00726-3>.
3. N. Elgiriyewithana, “Emotions,” Jan 2024. [Online]. Available: <https://www.kaggle.com/datasets/nelgiriyewithana/emotions>.
4. H. A. Sayyed, S. R. Sugave, S. Paygude, B. N. Jagdale, “Study and Analysis of Emotion Classification on Textual Data,” Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES-2021), 2021. Available: <https://doi.org/10.1109/ICCES51350.2021.9489204>.
5. I. S. Wibowo, A. Witanti, I. Susilawati, “Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 2024. Available: <https://jurnal.mdp.ac.id/index.php/jatisi/article/download/6718/1758>.
6. A. K. S. Santoso, A. Noviriandini, A. Kurniasih, “Klasifikasi Persepsi Pengguna Twitter Terhadap Kasus Covid-19 Menggunakan Metode Logistic Regression,” Jurnal Informatika, 2021. Available: <https://repository.bsi.ac.id/repo/files/374064/download/517-1143-1-PB-ok.pdf>
7. A. Defazio, F. Bach, S. Lacoste-Julien, “SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives, ” 2014. Available: <https://arxiv.org/abs/1407.0202>.